

# CAN 总线实时调度算法的应用

车进辉, 薛曼玲, 姜 丰

(北华大学计算机科学技术学院, 吉林 132021)

**摘要:**为满足电动车 CAN 通信系统在电动车能源管理系统中对实时性和动态分配带宽的要求, 提出一种基于事件通道的 CAN 总线实时调度算法。该算法利用后端通道配置数据库实时修改 CAN 总线通道组合方式, 为具有不同实时性和可靠性要求的数据动态分配传输通道, 满足电动车能源管理系统研究和实施阶段对系统可修改性的需求, 缩短系统的研发周期。

**关键词:** CAN 总线; 实时调度; 电动车

## Application of CAN Bus Real-time Scheduling Algorithm

CHE Jin-hui, XUE Man-ling, JIANG Feng

(College of Computer Science and Technology, Beihua University, Jilin 132021)

**【Abstract】** CAN bus real-time scheduling algorithm is proposed to satisfy the need for real-time and bandwidth dynamic allocation of CAN bus in electric vehicle. Backend channel configure database is used for modification of CAN bus channels at runtime. Requirements of modification in research and execute phases are meet properly by this algorithm. The algorithm shortens the time of the research and development time.

**【Key words】** CAN bus; real-time scheduling; electric vehicle

CAN 总线在汽车总线系统中得到越来越广泛的应用。随着石油等能源的日趋紧张, 混合动力汽车和纯电动汽车渐渐发展起来。动力电池是电动汽车的能源核心, 因此, 电动车能源管理在电动车中处于比较重要的地位。本文基于 TTCAN 思想<sup>[1]</sup>, 在固定优先级调度算法<sup>[2]</sup>和 EDF 算法<sup>[3]</sup>的基础上实现基于事件的总线实时调度算法。该算法为“863”计划电动车项目的能源管理部分提供了一个实时通信方案。

### 1 CAN 实时通道调度算法

#### 1.1 事件和事件通道

事件 E 定义为一个四元组

$E = \{\text{subject}, \text{context\_attribute\_list}, \text{quality\_attribute\_list}, \text{data}\}$

其中, subject 为事件标识; context\_attribute\_list 为事件的语境参数表, 内含事件发生时的上下文环境, 如事件发生的时间、位置; quality\_attribute\_list 为质量参数表, 内含事件的质量属性, 如截止时间、忽视容忍度; data 为事件相关数据, 如温度值、速度值。

事件通过事件通道从发布者传送给订阅者。事件通道是网络资源的抽象, 资源可以是物理网络中的频率、时隙等。一个事件通道是一个类型的实例, 当节点要分发事件或订阅事件时, 相应的通道被建立。EC 可用一个三元组定义

$EC = \{\text{subject}, \text{attribute\_list}, \text{handlers}\}$

其中, subject 是通道标识; attribute\_list 是通道属性表。

#### 1.2 通道配置数据库

通道配置数据库是 CAN 总线通道的配置库, 该数据库描述了网络资源和事件通道的资源分配。通道数据库 CCD 可定义为

$CCD = \{\{R\}, \{\text{distribute}(R, EC)\}, \{\text{FrameDisc}\}, \text{handle}\}$

其中, R 为网络资源集合; FrameDisc 是 E.data 数据结构的描述; distribute(R, EC) 是资源对事件通道的分配方法集合; handle 是 CCD 配置变更后接收变更通知的句柄。

#### 1.3 调度优先级和通道的划分

事件优先级划分为硬实时、软实时和非实时 3 个级别。硬实时事件不可超出截止时间调度, 一旦事件发生, 必须获得总线的占有权。软实时事件的截止时间大于 0 小于一个固定值, 在该固定值时间内, 事件如果没有获得总线的使用权, 则该事件已经没有意义。非实时消息在截止时间没有被调度, 但过期调度仍有一定意义。非实时事件只有在硬实时和软实时消息调度完毕总线空闲时才会被调度。根据事件的优先级, 通道类型也被划分成相应的 3 个级别: 硬实时通道用于硬实时事件的分发, 软实时通道用于分发软实时事件, 非实时通道用于分发非实时事件。

#### 1.4 总线时间同步和时隙的划分

通过广播时钟短帧实现 CAN 网时间同步。时钟帧具有硬实时优先级。该帧可以作为全局调度的时间起点。由于工程中 CAN 节点时钟存在差异, 因此时隙之间需要一个安全时间带, 如图 1 所示。这个时间带可以在配置数据库中根据工程需要进行动态调节。

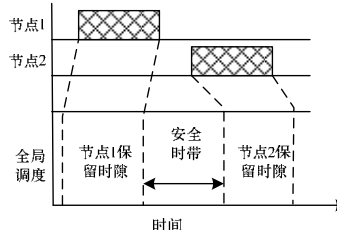


图 1 保留时隙间的安全时间带

**基金项目:** 国家“863”计划基金资助项目(2006AA11A151)

**作者简介:** 车进辉(1971-), 男, 硕士, 主研方向: 嵌入式系统, 数据挖掘; 薛曼玲, 硕士; 姜 丰, 教授

**收稿日期:** 2009-08-24 **E-mail:** chejinhui2008@126.com

设  $\max(T_{\text{clockoffset}})=\pm\Delta Oc$  为最大时钟偏移, 则最大时滞  $\max(T_{\text{skew}})=2\times\Delta Oc$ , GAP 必须满足:

$$T_{\text{gap}} \geq 2\times\Delta Oc$$

### 1.5 通道-时隙映射和截止时间-标识符映射

通道和截止时间是 CAN 协议的高层抽象, 应用中必须映射到 CAN 的链路层。通道到时隙的映射采用通道时隙映射表的方式实现, 见表 1,  $t_1$  和  $t_2$  时隙被映射到  $c_1$  通道,  $t_3$  被映射到  $c_2$  通道。截止时间到 CAN 帧标识符的映射采用调度周期-优先级映射方法, 设调度优先级为  $Sp$ , 截至时间为  $Td$ , 调度周期为  $Ts$ , 则  $Sp=Td/Ts$ 。

表 1 通道-时隙映射表

时隙( $ti$ )	通道( $ci$ )
$t_1$	$c_1$
$t_2$	$c_1$
$t_3$	$c_2$

### 1.6 实时调度算法

通道配置数据库(CCD)配置变更算法如下:

```

CAN_Realttime_Schedule_ReConfigure(){
CCD 端:
Broadcast_Global_Configure_Change_Begin_Message;
Broadcast2Nodes(CCD);
Broadcast_Global_Configure_Change_End_Message;
节点端:
Establish_Local_CCD;
Establish_Channels;
Establish_Resources_Channel_Map;
}

```

事件分派算法如下:

```

CAN_Event_Dispatch(){
for each events{
if(E.subject==HARD_REALTIME_EVENT)
put2HRChanel(E);
else if (E.subject==SOFT_REALTIME_EVENT)
put2SRChanel(E);
else
put2NRChanel(E);
}
}

```

通道调度算法如下:

```

CAN_Realttime_Schedule(){
Wait_for_until_global_clock_received{
If(TimeSlot==myNodeTimeslot){
LookforCCD;
If(timeslot== HARD_REALTIME_SLOT)
Schedule_HardRealTime_Event_to_Bus;
If(timeslot== SOFT_REALTIME_SLOT){
Decrease_each_Channel_Event_DeadLine;
If(E.deadline is smallest)
Schedule_SoftRealTime_Event_to_Bus;
else
Schedule_NoneRealTime_Event_to_Bus;
}
}
}
}

```

## 2 电动车能源管理

### 2.1 电动车能源系统结构

图 2 为能源系统的硬件结构, 上位机通过 CAN/USB 转换器与 CAN 总线连接。

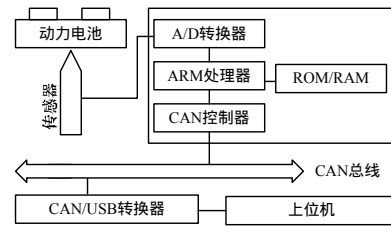


图 2 电动车能源系统硬件结构

图 3 为电动车能源系统的软件结构, 其中, CAN\_Realttime\_Schedule\_ReConfigure 模块的 CCD 端运行在上位机; 节点端运行在前端节点; CAN\_Event\_Dispatch 和 CAN\_Realttime\_Schedule 模块运行在前端节点。

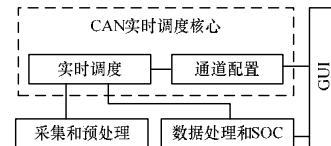


图 3 电动车能源系统软件结构

### 2.2 软件实现

前端软件开发和运行环境如下:

操作系统 ucLinux; 开发工具 ADS 1.2; 开发语言 C++。

后端软件开发和运行环境如下:

操作系统 Windows XP; 开发工具 LabView 7.1; 通道配置数据库在后端采用 XML 的格式, 经 LabView 解析后生成二进制配置文件广播给各个前端节点。

## 3 实验

实验主要测定单位时间事件发生数对调度算法的影响, 供工程参考使用。

实验环境如下:

实验采用三节点 CAN 总线, 其中, 2 个节点是采集前端; 另一个节点作为后端节点。时隙和通道划分为 4 时隙 2 通道, 其中一个时隙用于硬实时通道; 另一个时隙用于软实时通道。实验中 CAN 帧只携带实验所需时间数据, 其中,  $t_o$  为事件发生时间,  $t_q$  为事件入调度队列时间,  $t_s$  为事件被调度时间;  $t_r$  为事件到达目标节点时间, 调度延迟时间为  $t_d=t_r-t_s$ 。图 4 是事件调度的平均延迟时间与事件突发率之间的关系。事件突发率在每秒 1 000 次~4 000 次时, 调度延迟平稳地维持在 0.1 ms~0.15 ms。当事件突发率大于每秒 5 000 次时, 系统出现较大的拥塞, 调度延迟急剧增大。

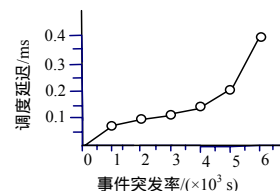


图 4 调度延迟曲线

## 4 结束语

基于事件和 EDF 的 CAN 总线调度算法在电动车能源系统中得到成功验证。结果表明, 调度算法可靠性高, 调度兼顾公平和效率。与 TTCAN 和事件调度算法相比, 具有较好的动态通道和时隙可配置特性, 可较好地满足系统开发前期和部署后对系统可修改性的需求。该算法已从项目中抽取出来形成组件, 并将在以后的项目中加入调度可行性分析部分。

(下转第 246 页)